IN THE CLAIMS:

1. (currently amended) A method for automatically validating text input, the method comprising:

~~an application~~ processing a markup language file~~, wherein the markup language file comprises~~ comprising a description of a graphical user interface (GUI), the description comprising a GUI element enabled to receive text input, wherein the markup language file comprises a markup language tag for instantiating a validation manager ~~component, wherein the description of the GUI comprises a description of a GUI element enabled to receive text input~~;

~~the application~~ instantiating the validation manager ~~component~~ in response to said processing the markup language file;

~~the application~~ displaying the GUI on a display screen of a computer system in response to said processing the markup language file;

~~providing~~ receiving text input to the GUI element enabled to receive text input;

~~the validation manager component~~ receiving a programmatic event by the validation manager in response to said ~~providing~~ received text input to the GUI element;

~~the validation manager component~~ determining whether the text input ~~provided to~~ received to the GUI element is valid text input in response to receiving the programmatic event; and

~~the validation manager component indicating~~ providing an indication that the text input ~~provided~~ received to the GUI element is invalid if the ~~validation manager component determines that the text input is not valid~~ the text input is determined to be invalid.

2. (currently amended) The method of claim 1, wherein the description of the GUI element enabled to receive text input comprises an attribute for specifying a type associated with the GUI element, the method further comprising:

the validation manager ~~component~~ instantiating a validation component, where said instantiating a validation component comprises specifying the type associated with the GUI element;

wherein said validation manager ~~component~~ determining whether the text input ~~provided to~~ at the GUI element is valid text input comprises the validation manager ~~component~~ calling the validation component;

wherein said validation manager ~~component~~ calling the validation component comprises the validation manager ~~component~~ specifying the text input ~~provided~~ to the GUI element;

wherein the validation component is operable to return a result value to the validation manager ~~component~~ indicating whether the text input ~~provided~~ received to the GUI element is valid text for the type associated with the GUI element.

3. (Currently amended) The method of claim 1, wherein

the step of receiving text input to the GUI element is performed ~~wherein said providing text input to the GUI element comprises~~ by a user of the application ~~providing text input to the GUI element~~.

4. (Currently amended) the method of claim 1,

wherein the step of receiving text input to the GUI element ~~said providing text in put to the GUI element~~ comprises an ~~the~~ application providing text input to the GUI element.

5. (Currently amended) the method of claim 1,

wherein the description of the GUI element enabled to receive text input comprises one or more attributes for specifying the text input to ~~the GUI element should~~ be validated; and

wherein the validation manager ~~component~~ is operable to validate text input to the GUI element in accordance with the one or more attributes specifying when text input to the GUI elements should be validated.

6. (Currently amended) the method of claim 5,

wherein each of the one or more attributes for specifying when text input to the GUI element should be validated corresponds to at least one type of programmatic event; and

wherein the step of said validation manager ~~component~~ receiving a programmatic event comprises the validation manager ~~component~~ ignoring the programmatic event if the programmatic event does not correspond to one of the attributes for specifying when text input to the GUI element should be validated.

7. (Currently amended) the method of claim 1,

wherein the step of receiving ~~said providing~~ text input to the GUI element comprises receiving ~~performing~~ one or more of the following actions ~~from the group consisting of~~:

changing the value of the GUI element,

pressing a key,

releasing a key,

causing the GUI element to gain user interface focus[[.]],

causing the GUI element to lose user interface focus,

moving a mouse pointer over the GUI element,

clicking on the GUI element,

double-clicking on the GUI element;

wherein said validation manager ~~component~~ receiving a programmatic event in response to said ~~providing~~ text input to the GUI element comprises the validation manager ~~component~~ receiving a programmatic event corresponding to the action performed.

8. (Currently amended) The method of claim 1,

wherein the markup language tag for instantiating the validation manager ~~component~~ comprises one or more parameters for specifying the default behavior of when the validation manager ~~component~~ should validate text input for GUI elements described in the markup language file.

9. (Currently amended) The method of claim 8,

wherein the description of the GUI element enabled to receive text input comprises one or more attributes for specifying when text input to the GUI element should be validated; and

wherein the validation manager ~~component~~ is operable to override the default behavior and validate text input to the GUI element in accordance with the one or more attributes specifying when text input received to the GUI element should be validated.

10. (Currently amended) The method of claim 1,

wherein said validation manger ~~component~~ indicating that the text input ~~provided~~ received to the GUI element is invalid comprises the validation manager requesting the application to alter the visual appearance of the GUI element.

11. (Currently amended) The method of claim 1,

wherein said validation manager ~~component~~ indicating that the text input ~~provided~~ received to the GUI element is invalid comprises the validation manger displaying an informational user interface window.

12. (Currently amended) The method of claim1,

wherein the description of the GUI element enabled to receive text input comprises one or more attributes for controlling text input validation for the GUI element;

wherein said step of ~~application~~ processing a markup language file comprises the application constructing a document object representing the markup language file;

wherein ~~said application~~ instantiating the validation manager ~~component~~ comprises the application passing a reference to the document object to the validation manager ~~component~~; and

wherein, in response to being instantiated by and receiving the reference to the document object ~~from the application~~, the validation manger ~~component~~ is operable to traverse the document object in order to discover the one or more attributes for controlling text input validation for the GUI element.

13. (Original) The method of claim 12,

wherein the one or more attributes for controlling text input validation for the GUI element include an attribute for specifying a type associated with the GUI element.

14. (Original) The method of claim 12,

wherein the one or more attributes for controlling text input validation for the GUI element include one or more attributes for specifying when text input to the GUI element should be validated.

15. (Original) The method of claim 12,

wherein the one or more attributes for controlling text input validation for the GUI element include one or more attributes for specifying how invalid text input for the GUI element should be indicated.

16. (Currently amended) The method of claim 1,

wherein the validation manager ~~component~~ is a COM object.

17. (Currently amended) The method of claim 1,

wherein the validation manager ~~component~~ is a Java object.

18. (Original) The method of claim 1,

wherein the markup language is HTML.

19. (Currently amended) The method of claim 2, wherein the type associated with the GUI element is a type comprising one or more ~~from the group consisting~~ of:

Coordination of Benefits (COB) code, Current Procedural Terminology (CPT) code, HCFA Common Procedure Coding System (HCPCS) code, International Classification of Diseases (ICD) code, U.S. Employer Information Number (EIN), U.S. Social Security Number, currency, U.S. states and territories, telephone number, zip code, date, and boolean.